

Un errore di segmentazione si verifica quando un programma cerca di accedere a una zona di memoria non consentita oppure non allocata.

```
kali@kali: ~/Desktop
File Actions Edit View Help

(kali@kali)-[~]
$ cd /home/kali/Desktop

(kali@kali)-[~/Desktop]
$ nano BOF.c
```

```
GNU nano 8.1 BOF.c *
#include <stdio.h>

int main(){
    char buffer [10];

    printf ("Si prega di inserire il nome utente:");
    scanf ("%s", buffer);

    printf ("Nome utente inserito: %s\n", buffer);

    return 0;
}
```

```
$ nano BOF.c

(kali@kali)-[~/Desktop]
$ gcc -g BOF.c -o BOF

(kali@kali)-[~/Desktop]
$ ./BOF
Si prega di inserire il nome utente:Lilly
Nome utente inserito: Lilly

(kali@kali)-[~/Desktop]
$
```

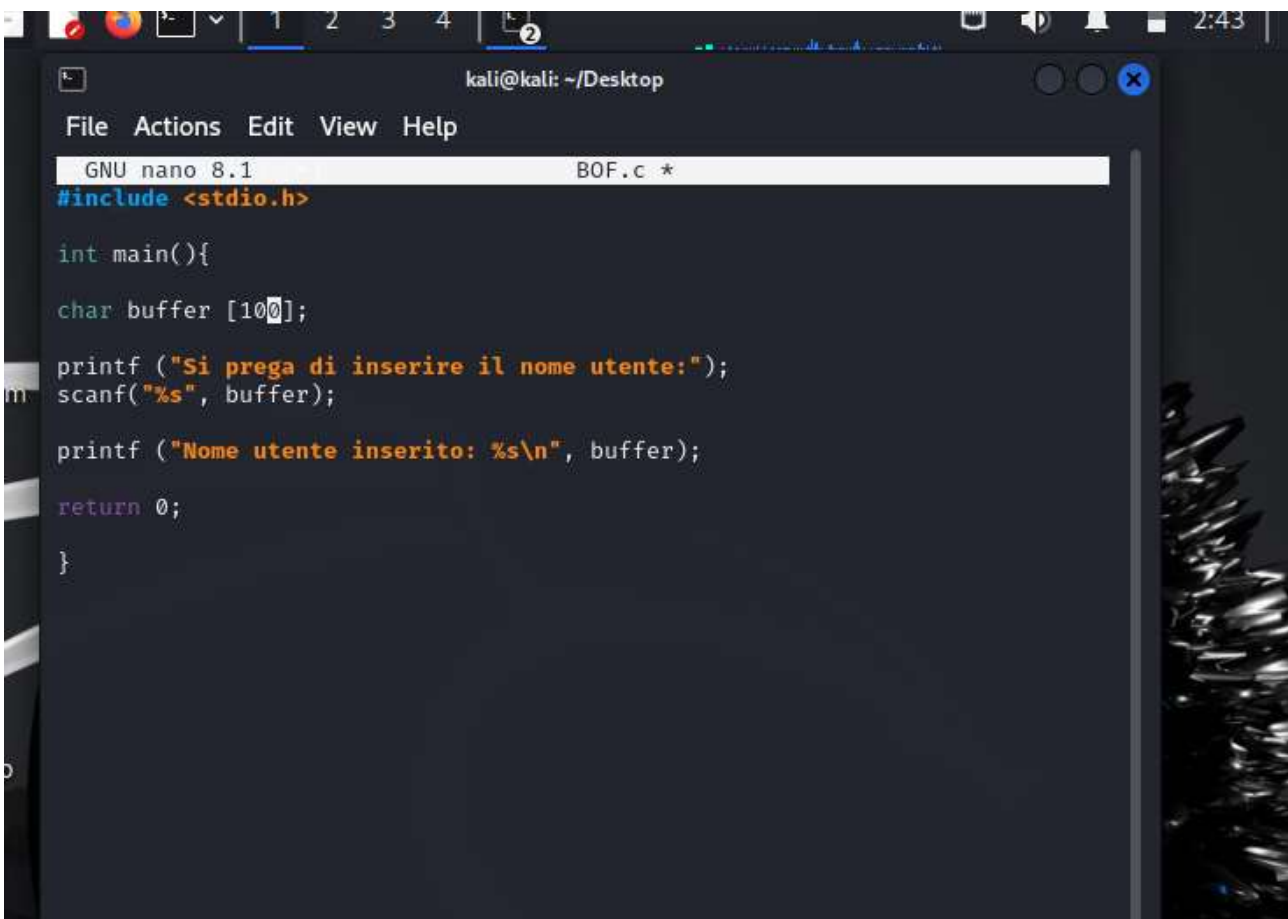
Avendo inserito come limite 10 caratteri quando viene inserito un nome utente più lungo viene

restituito errore overflow.

```
(kali@kali)-[~/Desktop]
$ ./BOF
Si prega di inserire il nome utente:Supercalifragilistichepsipiralidoso
Nome utente inserito: Supercalifragilistichepsipiralidoso
zsh: segmentation fault ./BOF

(kali@kali)-[~/Desktop]
$
```

Modificano il file BOF.c



```
kali@kali: ~/Desktop
File Actions Edit View Help
GNU nano 8.1 BOF.c *
#include <stdio.h>

int main(){
char buffer [100];

printf ("Si prega di inserire il nome utente:");
scanf ("%s", buffer);

printf ("Nome utente inserito: %s\n", buffer);

return 0;
}
```

```
25th. Segmentation Fault ./BOF
(kali@kali)-[~/Desktop]
$ nano BOF.c
(kali@kali)-[~/Desktop]
$ gcc -g BOF.c -o BOF
(kali@kali)-[~/Desktop]
$ ./BOF
Si prega di inserire il nome utente:Supercalifragilistichepsipiralidoso
Nome utente inserito: Supercalifragilistichepsipiralidoso
(kali@kali)-[~/Desktop]
$
```

Aumentare la dimensione dell'array potrebbe ridurre il rischio di un errore di buffer overflow (BOF), ma non elimina automaticamente il rischio di segmentazione.

```
File  Actions  Edit  View  Help
GNU nano 8.1      BOF.c
#include <stdio.h>
#include <stdlib.h>

int main(){

char buffer [30];

printf ("Si prega di inserire il nome utente:");
if (scanf("%29s", buffer) !=1){
printf ("Errore input");
return 1;
}
printf ("Nome utente inserito: %s\n", buffer);

return 0;

}
```